# 16.7M Pixel 8000fps Sparse Binarized Scientific Image Sensor

**Peng Gao**, Sampsa Veijalainen, Jente Basteleus, Gaozhan Cai, Bert Luyssaert, Bart Dierickx

*Caeleste, Belgium*

*IISW 2019*

# Outline

- Introduction
- Challenges and design solutions
- Measurement results
- Take home message

# Introduction

## Binarized sensor for counting based image

- Application: sparse particle detection

## Key Features

- High frame rate
  - Higher flux
  - Less motion blur
- Low variability detection threshold:
  - Lower bit error rate

# Summary of performance

| | |
|---|---|
| Pixel Array | 8 µm 4T 4096 x 4096<br>4k by 6k, 4k by 8k are possible stitch configurations |
| Frame rate | 8000 fps |
| Equivalent Pixel rate | 8kfps x 4k x 4k = 134 Gpixel/s |
| Output interface | 48 Gbit/s (64channels*750Mbit/s) |
| Data compression | Yes |
| Shutter type | Rolling readout |
| ADC | 1 bit |
| CVF at FD (before ADC) | 66 µV/e- (580 µV/e-) |
| Bit error rate | 1e-6 |
| Min detection threshold (σ) | 130e- (28e-$_{RMS}$) |
| Power | <5W @ 3.3V & 1.8V |
| Technology | CIS 0.18µm |

# Outline

- Introduction
- Challenges and design solutions
- Measurement results
- Take home message

# Frame rate challenge: 8000fps

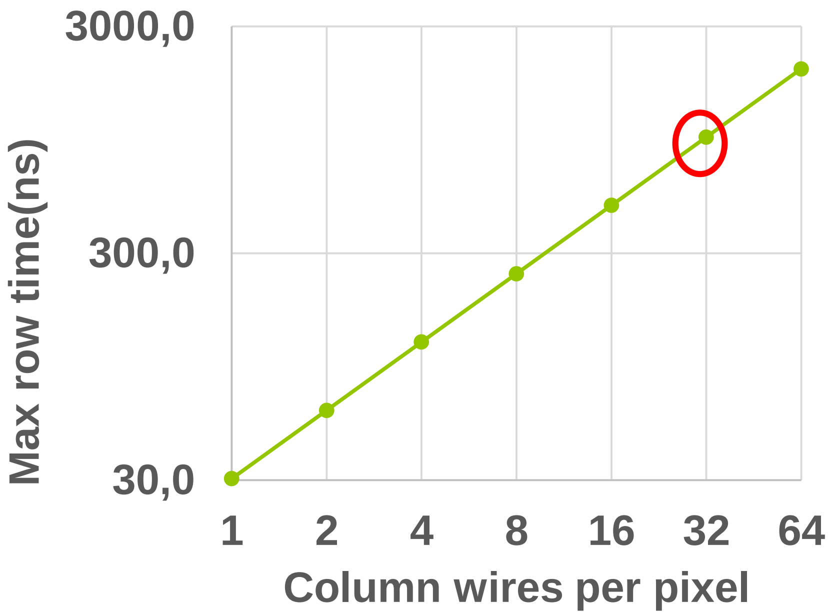
caeleste

## Long wire settling:

- ➢ Source follower and 4cm wire
  - RC $\tau >$ **200ns**
  - Available Row time $\mathbf{T_{1row}} \approx \boldsymbol{30ns}$
- ➢ Long video line

## Solution

- Parallelism
- Segmentation
- Current readout instead of voltage



## Implementation

- 32 rows readout in parallel. $\mathbf{T_{32row}} \approx \boldsymbol{1\mu s}$
- North, South separation: column wire RC 4X
- Column segmentation
- TIA (Transimpedance amplifier) in both column and video line readout

# Data rate challenge

## If all pixels would be read at 8000fps:

- 134 Gpixel/s
- Or after binarization: **134Gbit/s**

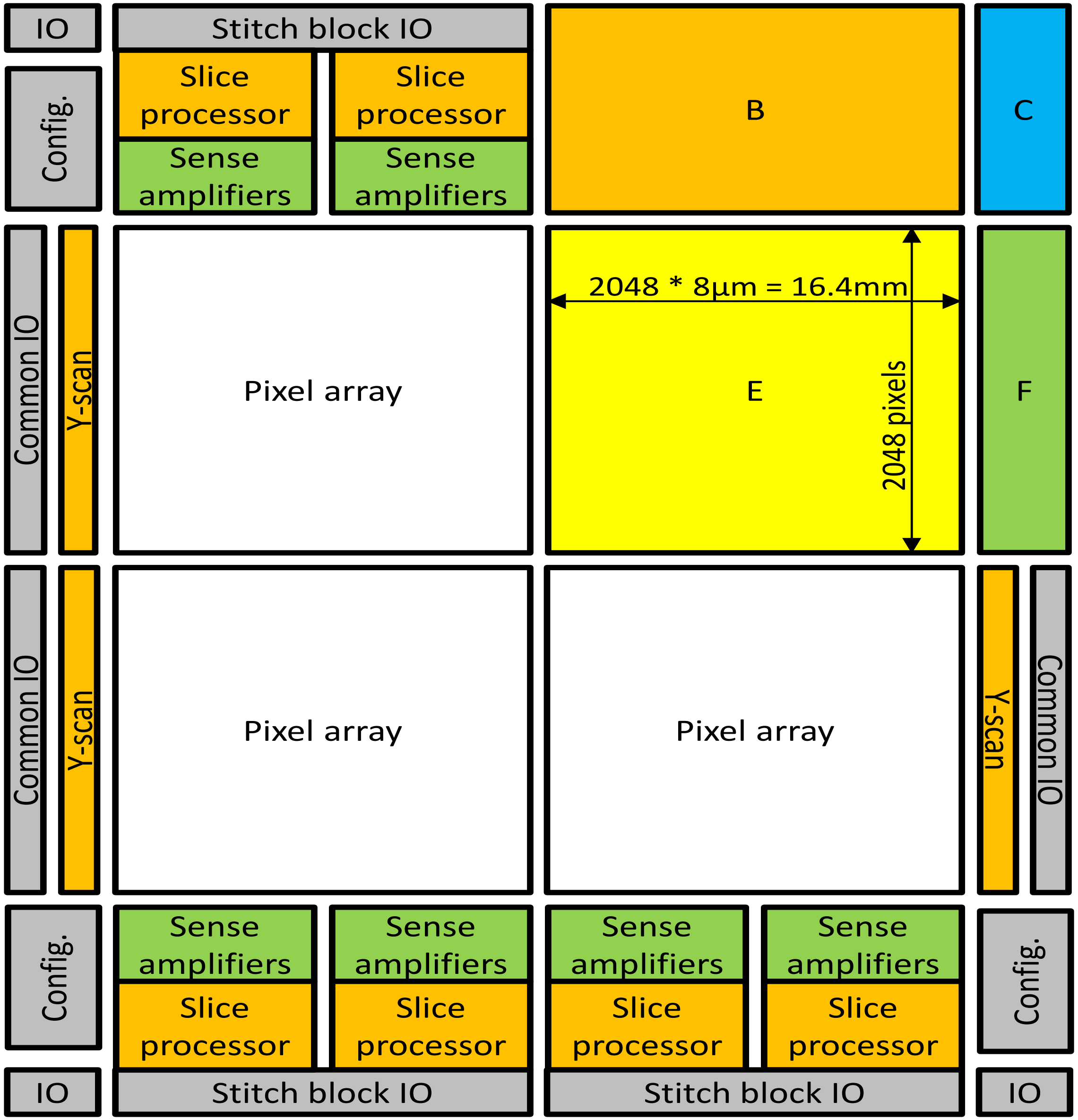
## Solution

- On chip kernel (group of pixel) based sparse image compression
- Only the kernels that contain (at least) one hit are read out

## Implementation

- “slices” of pixels = 2 times 16 rows are copied to processing area
  - Each side 16 rows
- Subdivided into 4x16 pixels kernels and coded unique address
- Kernels are evaluated and scanned by a priority encoder
- 64 bits data and 8 bits address are output over 64 parallel LVDS channels at 750Mbit/s/ch. i.e. 48Gbit/s
- ~32% of the array

# Architecture



IO | Stitch block IO | B | C

Config. | Slice processor | Slice processor

Sense amplifiers | Sense amplifiers

Common IO | Y-scan | Pixel array | E | F

2048 * 8µm = 16.4mm

2048 pixels

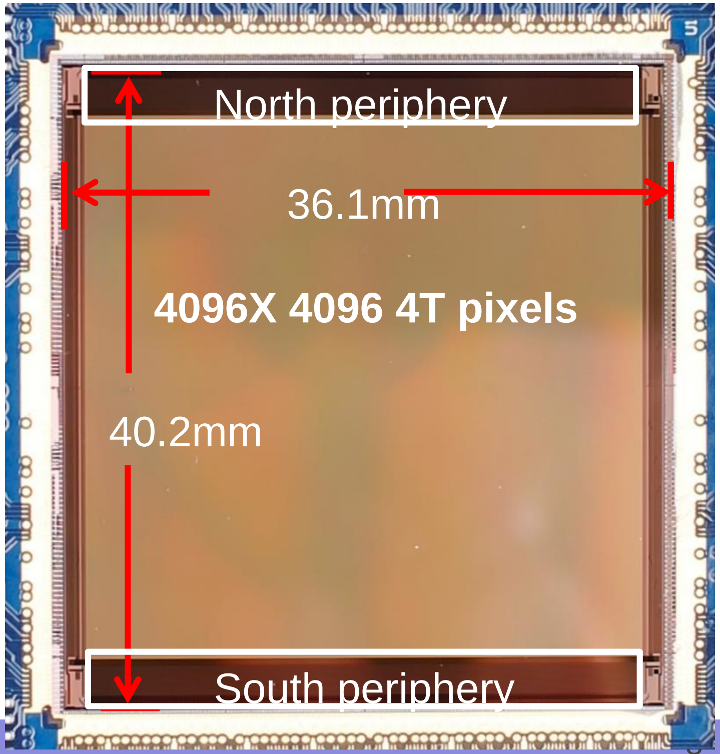
Common IO | Y-scan | Pixel array | Pixel array | Y-scan | Common IO

Config. | Sense amplifiers | Sense amplifiers | Sense amplifiers | Sense amplifiers | Config.

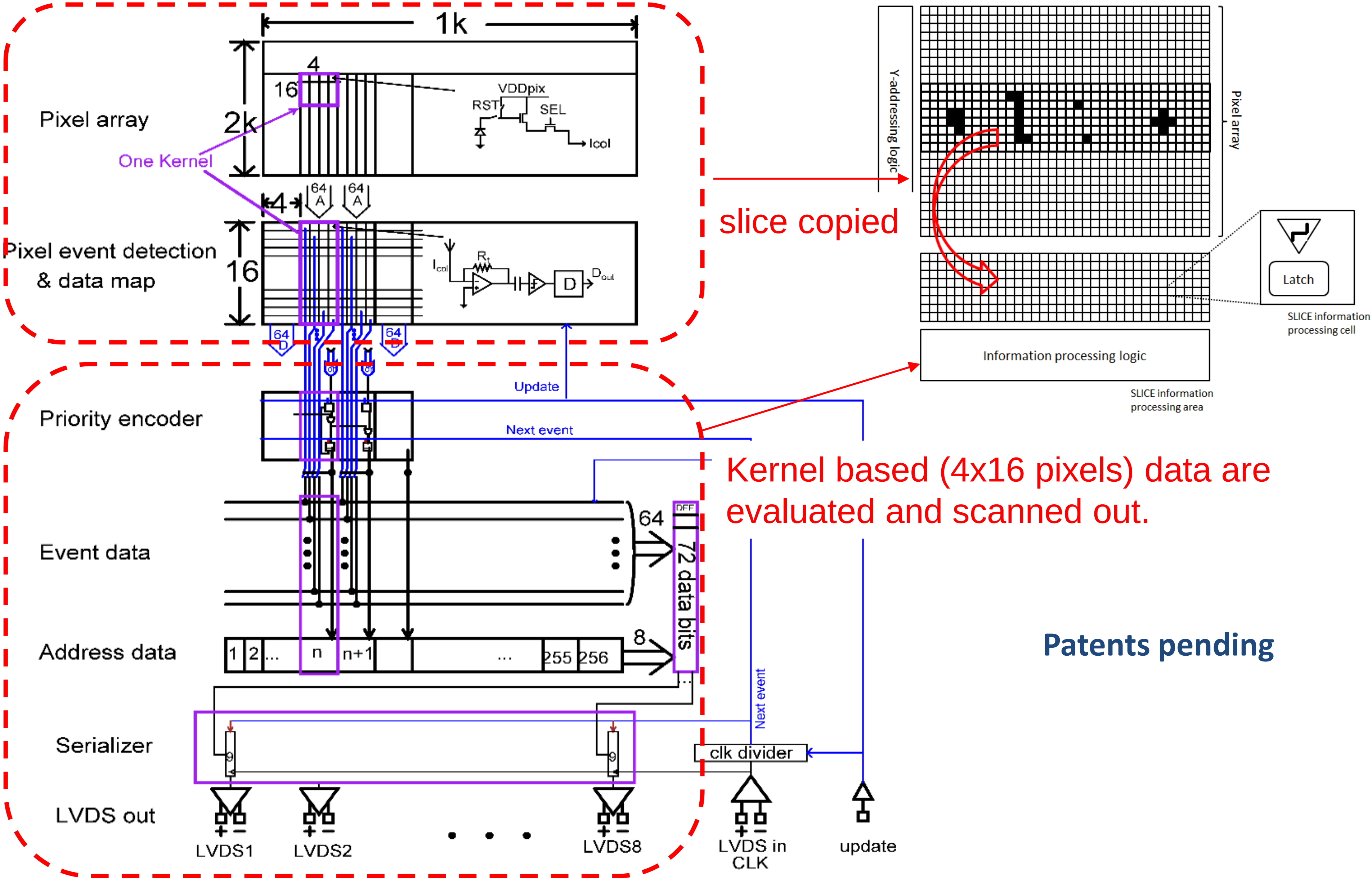
Slice processor | Slice processor | Slice processor | Slice processor

IO | Stitch block IO | Stitch block IO | IO

- **2 by 2 stitch**
- **2 side read readout**
- **Simultaneously 16 rows readout at each side**
- **Segmentation: 4 X 1024 columns**



# Signal chain of one segment



slice copied

Kernel based (4x16 pixels) data are evaluated and scanned out.

**Patents pending**

# Detection threshold challenge

caeleste

- Particles create charge packets of 200… 1000 secondary electrons, collected by one or multiple photodiode
- Low bit error rate: < 1e-6
- Threshold : <150 e-
  - With low static variability
  - With low temporal variability (noise)

# Detection threshold variability

Solutions:

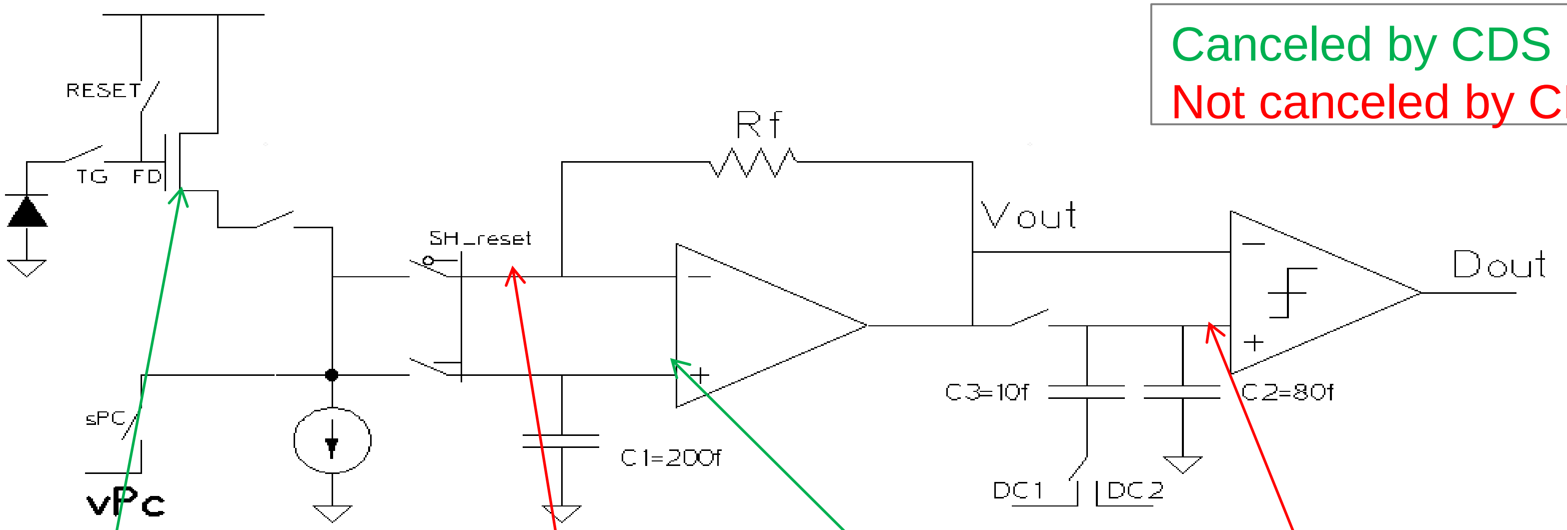
- CDS
- High signal gain ( $CVF_{TIAout}$ ) before CDS and comparator

$$CVF_{TIAout} = CVF_{FD} * g_{msf} * R_f$$

- Remaining mismatch and noise: device sizing

The variability of detection threshold (σ) < 28e-

Canceled by CDS
Not canceled by CDS



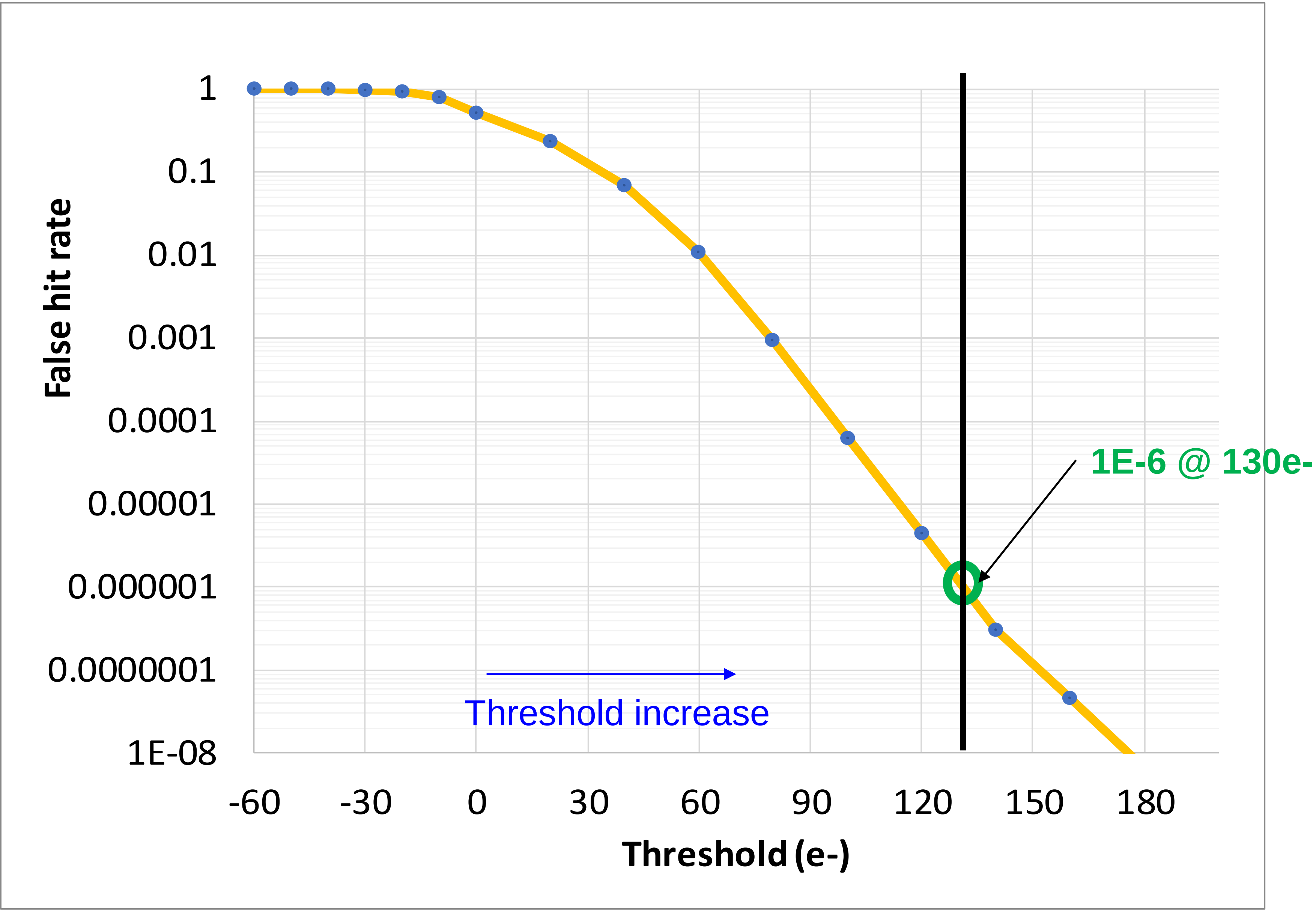
*Variability contributors*

$$\sigma_{V_{OS_{sf}}} + KTC_{FD} + \varphi * \frac{\sigma_{CVF_{TIAout}}}{CVF_{TIA_{out}}} + \frac{KTC_{C1} + \sigma_{V_{OS_{TIA}}}}{A_{SF} * CVF_{FD}} + \frac{KTC_{C23} + \sigma_{V_{OS_{comp}}}}{CVF_{TIA_{out}}}$$

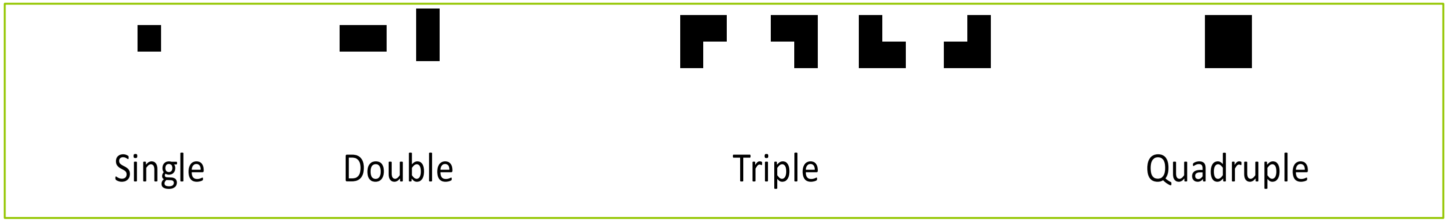
# Outline

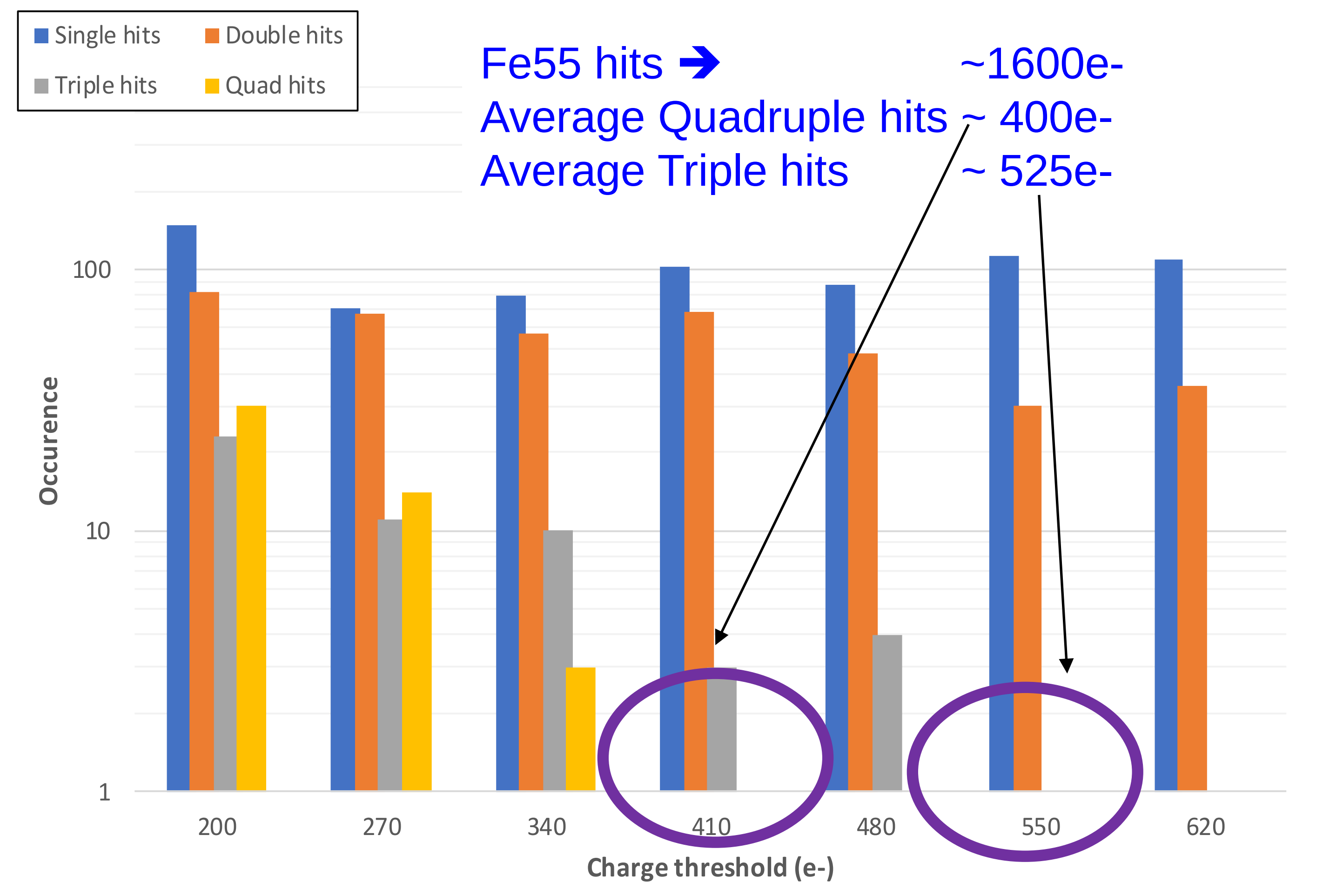
- Introduction
- Challenges and design solutions
- Measurement results
- Take home message

# Threshold vs false hit rate



# Fe55 test result





Fe55 hits → ~1600e-
Average Quadruple hits ~ 400e-
Average Triple hits ~ 525e-

# Outline

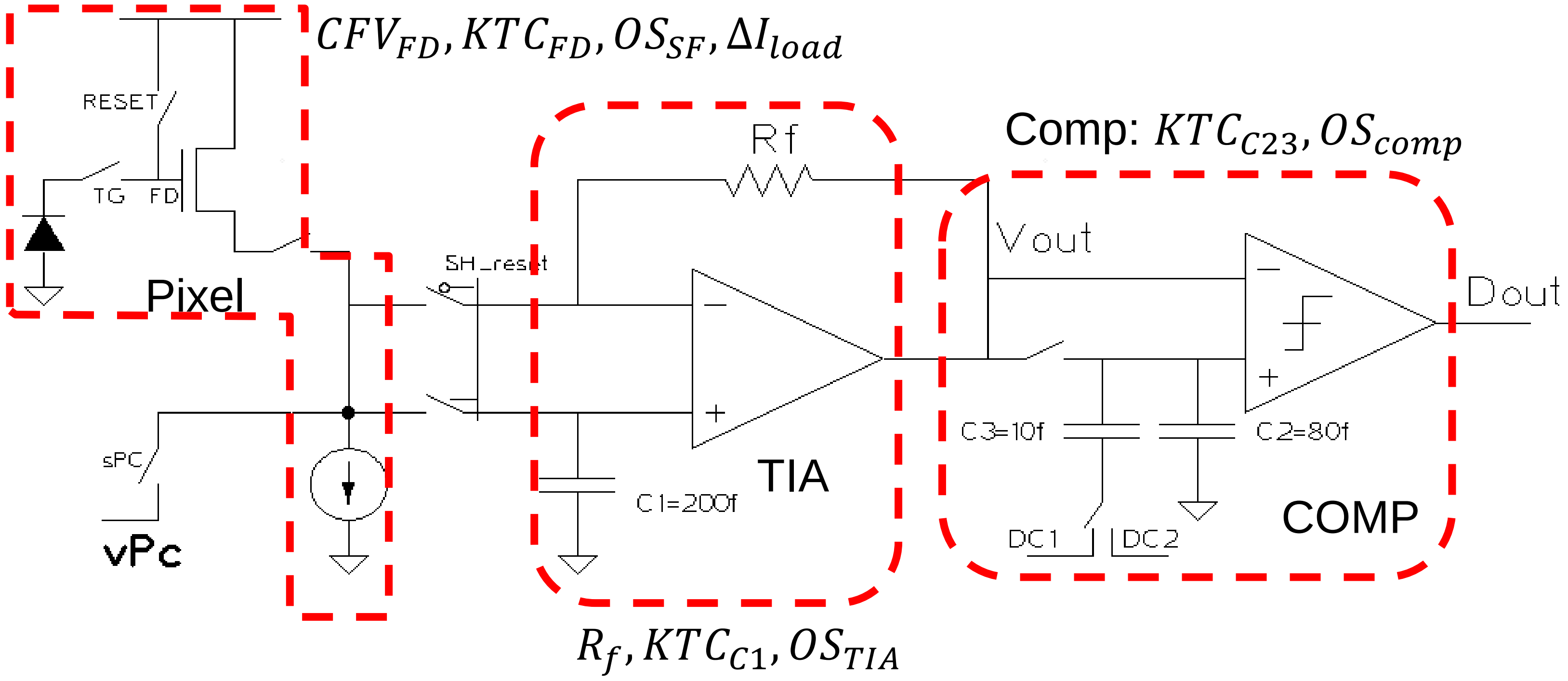
- Introduction
- Challenges and solutions
- Measurement results
- Take home message

# Take home messages

- **Large array high frame rate achieved by:**
  - Exploiting sparseness
  - Current mode read out to speed up settling
- **Low detection threshold realized by**
  - CDS
  - High signal path gain prior to CDS and comparator
  - Device sizing for matching and low noise

# Thank you!
# Questions?

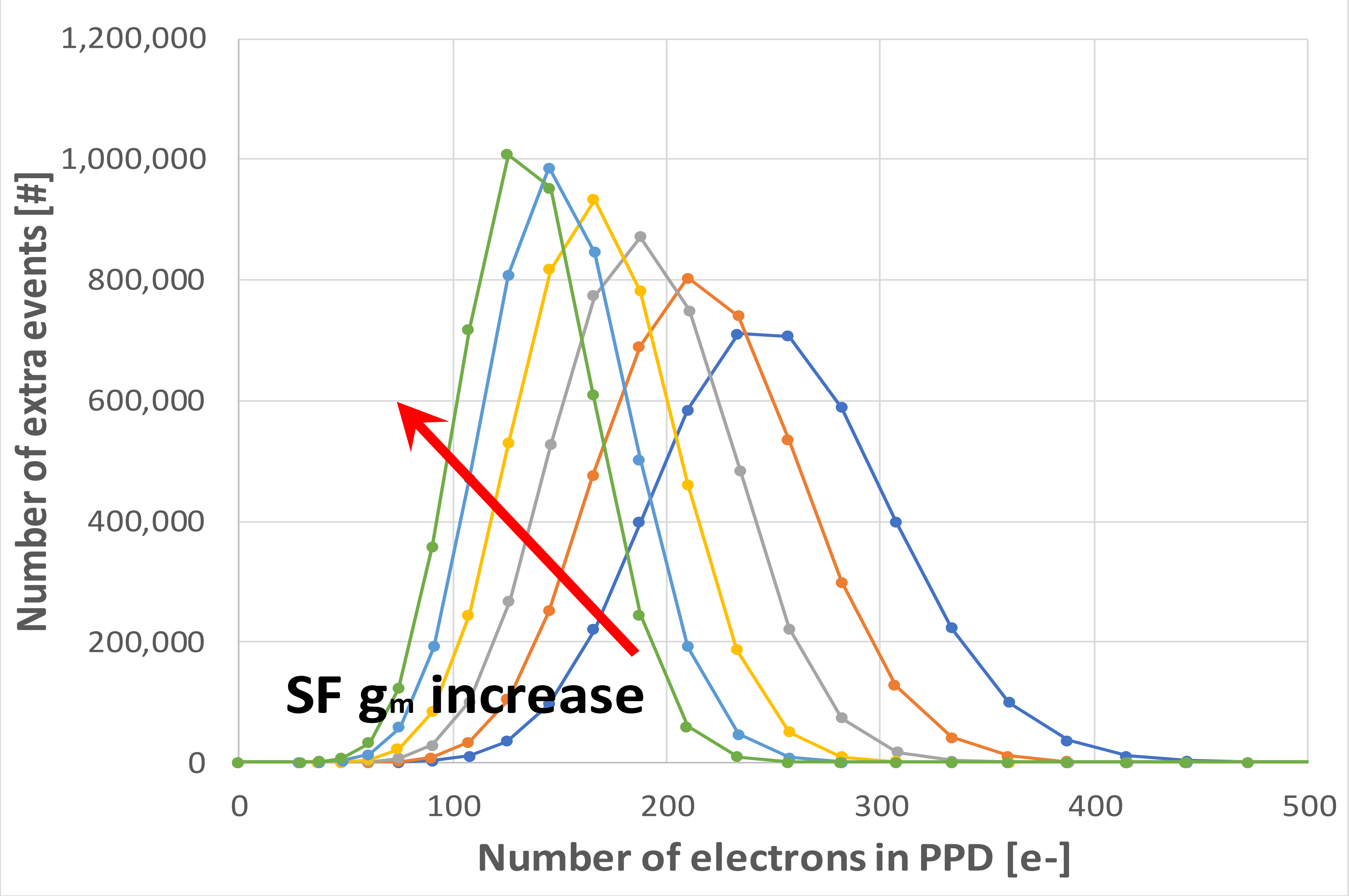
# Detection threshold variability



$CFV_{FD}, KTC_{FD}, OS_{SF}, \Delta I_{load}$

Comp: $KTC_{C23}, OS_{comp}$

$R_f, KTC_{C1}, OS_{TIA}$


1,200,000
1,000,000
800,000
600,000
400,000
200,000
0
Number of extra events [#]
0
100
200
300
400
500
Number of electrons in PPD [e-]
SF $g_m$ increase

# Data rate challenge

## If all pixels would be read at 8000fps:

- 134 Gpixel/s
- Or after binarization: **134Gbit/s**

## Solution

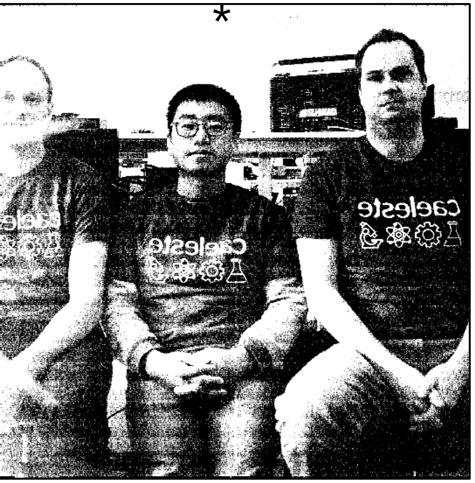
- On chip kernel (group of pixel) based sparse image compression
- Only the kernels that contain (at least) one hit are read out

## Implementation

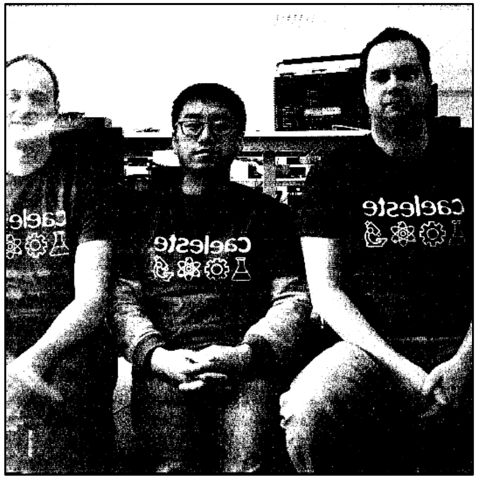
- “slices” of pixels = 2 times 16 rows are copied to processing area
  - Each side 16 rows
- Subdivided into 4x16 pixels kernels and coded unique address
- Kernels are evaluated and scanned by a priority encoder
- Kernel data (64 bits) and address (8 bits) are output over 64 parallel LVDS channels at 750Mbit/s/ch.  i.e. 48Gbit/s
- ~32% of the array

# Threshold binarized image

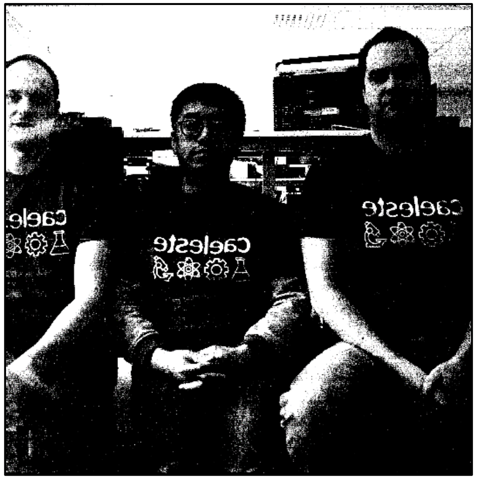
* This is not the real application.  Just a demonstration of binarized images *



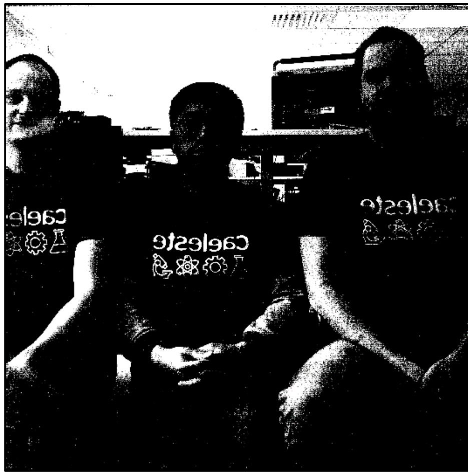
Threshold 100 e-



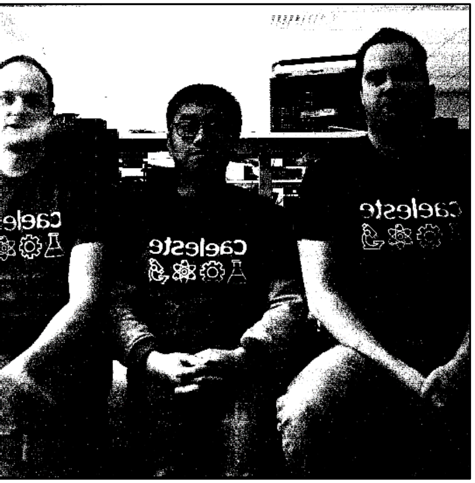
Threshold 140 e-



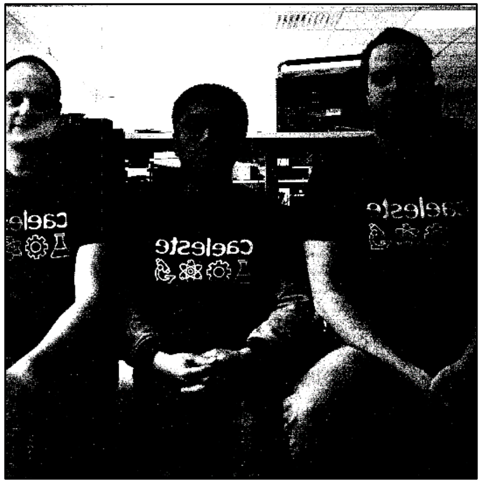
Threshold 180 e-



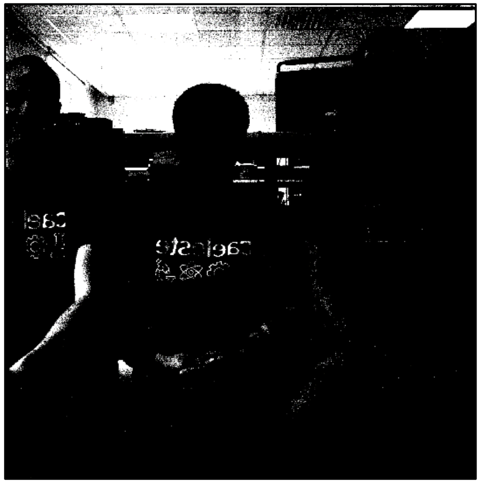
Threshold 220 e-



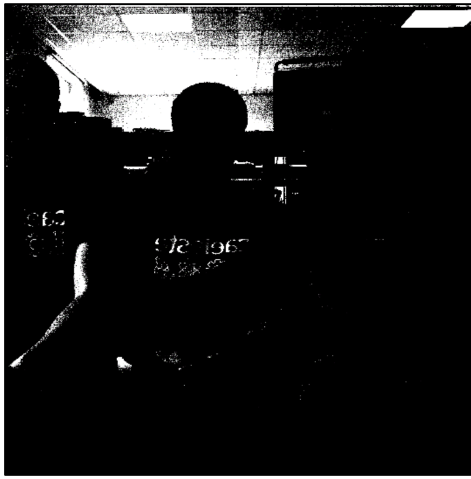
Threshold 260 e-



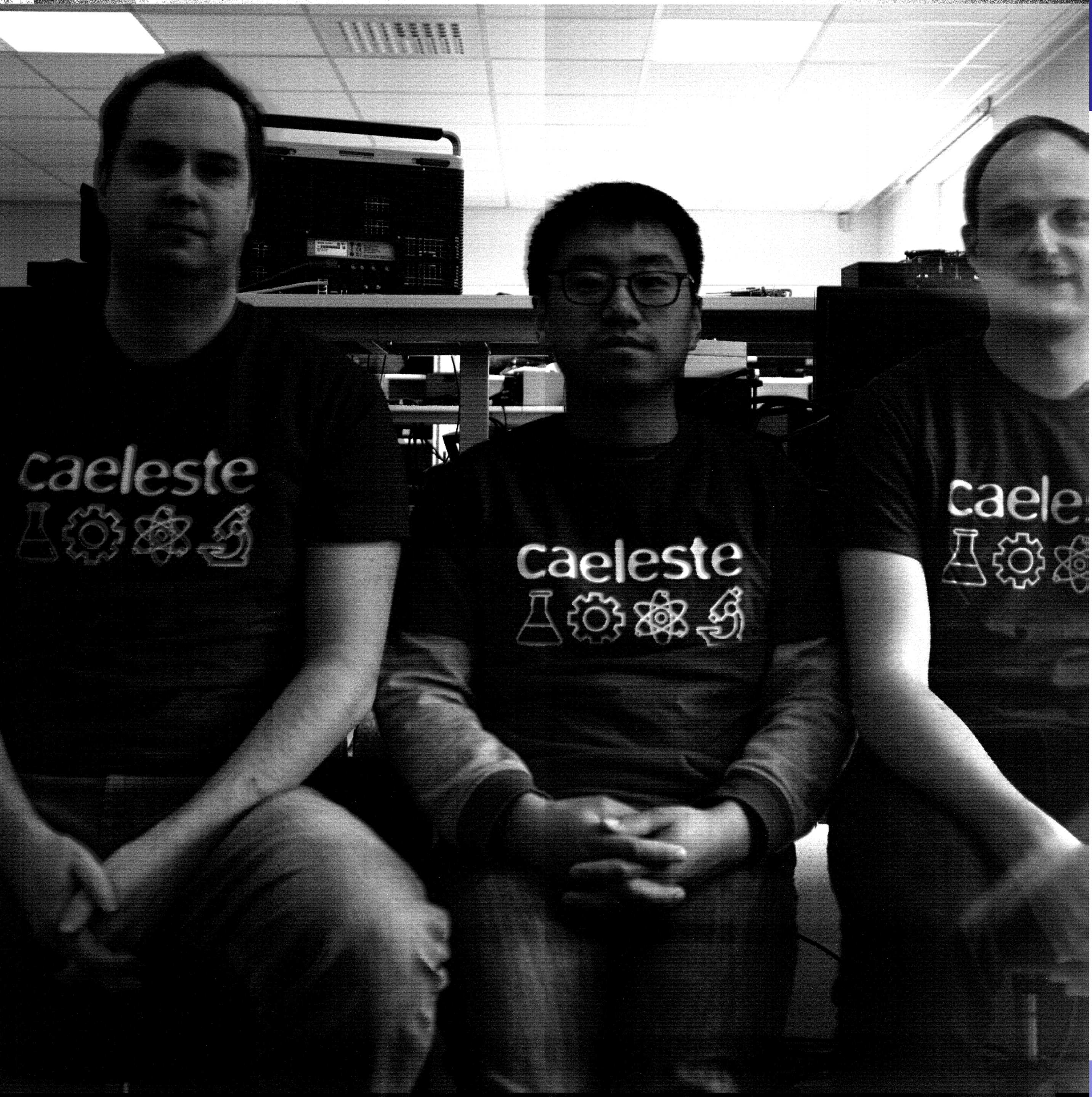
Threshold 300 e-



Threshold 340 e-



Threshold 380 e-



Average of the previous 8 binarized images
➔ a 3-bit image